Chaos

Detecting unstable periodic orbits based only on time series: When adaptive delayed feedback control meets reservoir computing

Cite as: Chaos **29**, 093125 (2019); https://doi.org/10.1063/1.5120867
Submitted: 22 July 2019 . Accepted: 28 August 2019 . Published Online: 24 September 2019

Qunxi Zhu, Huanfei Ma, and Wei Lin

Detecting unstable periodic orbits based only on time series: When adaptive delayed feedback control meets reservoir computing

Cite as: Chaos **29**, 093125 (2019); doi: 10.1063/1.5120867
Submitted: 22 July 2019 · Accepted: 28 August 2019 ·
Published Online: 24 September 2019

Qunxi Zhu,[1,2] Huanfei Ma,[3] and Wei Lin[1,2,4,a)]

AFFILIATIONS

[1]School of Mathematical Sciences, Fudan University, Shanghai 200433, China
[2]Shanghai Key Laboratory for Contemporary Applied Mathematics, LMNS (Fudan University), and LCNBI (Fudan University), Shanghai 200433, China
[3]School of Mathematical Sciences, Soochow University, Suzhou 215006, China
[4]Centre for Computational Systems Biology, Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai 200433, China

Note: This paper is part of the Focus Issue, "When Machine Learning Meets Complex Systems: Networks, Chaos and Nonlinear Dynamics."
[a)]**Electronic mail:** wlin@fudan.edu.cn

ABSTRACT

In this article, we focus on a topic of detecting unstable periodic orbits (UPOs) only based on the time series observed from the nonlinear dynamical system whose explicit model is completely unknown *a priori*. We articulate a data-driven and model-free method which connects a well-known machine learning technique, the reservoir computing, with a widely-used control strategy of nonlinear dynamical systems, the adaptive delayed feedback control. We demonstrate the advantages and effectiveness of the articulated method through detecting and controlling UPOs in representative examples and also show how those configurations of the reservoir computing in our method influence the accuracy of UPOs detection. Additionally and more interestingly, from the viewpoint of synchronization, we analytically and numerically illustrate the effectiveness of the reservoir computing in dynamical systems learning and prediction.

With tremendous development of data collections, processing, and mining, data-driven and model-free methods and algorithms are highly expected in various communities of science, engineering, and even social science. Those complex data, experimentally collected along with the time evolution, are usually supposed to be produced by the hidden complex dynamical systems, whose explicit models are often unknown partially or even completely. In order to depict the skeleton of such complex data and make accurate prediction without any explicit models, advanced technologies of machine learning, such as the artificial neural networks and the reservoir computing, have been invited and applied successfully. This article proposes an integrated framework connecting the reservoir computing with the adaptive delayed feedback control and demonstrates the effectiveness of the proposed framework in locating and controlling the unobservable and unstable periodic orbits based merely on the collected data of chaotic time series produced by complex systems. More interestingly and significantly, this article also presents an analytical argument from a viewpoint of dynamical synchronization to illustrate the effectiveness of the reservoir computing in dynamical systems learning and prediction. It is anticipated that our works including the machine learning techniques are able to meet the requirements arising in the research of complex systems, and also that the theory of complex dynamical systems can inversely contribute to illustrating and complementing the principles of machine learning techniques.

I. INTRODUCTION

Controlling chaos in the nonlinear systems has been investigated for about three decades since the seminal work contributed

by Ott, Grebogi, and Yorke (OGY).[1] Particularly, they demonstrated the effectiveness of a proposed technique of mild external force[1] to stabilize the unstable periodic orbits (UPOs) that are embedded in the underlying chaotic attractor (invariant set) and referred to the skeleton of classical and even quantum chaos.[2] In addition to this OGY technique, many novel methods have been thereby proposed in the vast literature for realizing such stabilization and modulating the corresponding complexity in various physical, biological, chemical, and engineering systems.[3–11] For example, the delayed feedback control law, a noninvasive control technique, was introduced in Ref. 12 and consequently extended for achieving UPOs stabilization, and the celebrated Gutzwiller trace formula was established, describing a quantum-classical relation between the quantum energy levels and the classical periodic orbits.[13] The investigation of UPOs has played significant roles in the rapid development of nonlinear dynamics and nonlinear sciences.

In the experimental study of nonlinear dynamics, a lot of time series, including the data of chemical reaction systems and the biophysical signals of brains, are measured from real systems whose vector field equations are partially or completely unknown. It therefore invites an urgent problem of detecting or extracting UPOs without knowing the specific equations of the interested systems but only with the measured time series. So far, there has been some outstanding bottom-up studies on this topic.[14–18] Some of these studies usually begin with utilization of the delay-coordinate embedding technique[19] for reconstructing the underlying dynamical system in the sense of one-to-one mapping, and then design some approach, such as the one proposed in Ref. 15, to detect the UPOs from the only short segments of available time series. Some detect UPOs statistically by utilizing the linear dynamics around a UPO to produce a statistical measure which, in theory, is singular at the UPO.[17,18] In spite of all their usefulness in applications, such methods sometimes are limited either to the systems of discrete-time or/and low-dimension, or to the insufficient information of linear dynamics around the concerned UPOs. Thus, highly expected is a data-driven and model-free method for accurately locating UPOs embedded in the unknown systems of continuous time, high dimension, or/and even infinite dimension.

Recently, there have been a tremendous deal of interest in applying machine learning technologies to the fields replete with the measured data of multimodality. Successful applications abound, including image recognition,[20] speech recognition,[21] playing Go,[22] and biostructure identification.[23] In addition to these applications based mostly on the traditional frameworks of convolutional neural networks and recurrent neural networks, focal topics also include utilization of machine learning technology, such as the compressed sensing[24–26] and the reservoir computing (RC),[27–31] to model or predict the complex dynamics of nonlinear systems using the observed data of time series.[32–37] Among all the practical technologies, the RC is a representative computational framework that could be regarded as an extension of recurrent neural networks.[38] Typically, it contains a high- or infinite-dimensional system,[39,40] independently of the considered task and called a "reservoir." The reservoir is driven by an input signal and maps the input space to the "feature space." Then, a simple "readout" mechanism or a postprocessing function (usually determined by a linear regression) is trained to read the extracted features and map it to the expected output. Notably, the training process is only performed during the postprocessing (readout) and the feature extracting process (reservoir) is fixed all the time. Additionally, the RC framework has been physically implemented in the hardware designs,[39–41] which further confirms its great potential in real applications.

Thereby, connecting the machine learning to the UPOs detection based only on the data of time series becomes the theme of this article. As a matter of fact, the idea of connecting these two topics have been developed initially in Ref. 42, where the traditional neural networks are used to adaptively restore the vector fields of the interested nonlinear systems and the delay-coordinate embedding technique[19] is used to reconstruct the appropriate coordinate system of phase space as partial internal states are available for observation. In this article, we are to use the RC framework instead to restore the vector fields, which, in principle, does not require the embedding technique for the observation that only includes partial internal states. We will then employ the adaptive delayed feedback control (ADFC)[6–8] to the systems restored by the RC framework for detecting the UPOs. We will show how the RC configurations in time series training influence the accuracy of UPOs detection. We, therefore, aim to articulate a data-driven and model-free method for detecting the UPOs based only on the time series observed from the unknown systems. Additionally but more interestingly, we intend to present an analytical argument from a viewpoint of synchronization to illustrate the effectiveness of the RC framework in dynamical systems learning and prediction.

The rest of this article is organized as follows. Section II introduces the RC framework, the ADFC technique, and their integrated model for UPOs detection. Section III provides several representative examples to demonstrate the effectiveness of the articulated method. Section IV illustrates the effectiveness of the RC framework from a viewpoint of synchronization. Section V closes this article by presenting some conclusions and perspectives.

II. MODEL FORMULATIONS: INTEGRATING RC WITH ADFC

A. Restoration of dynamical systems using RC framework

To begin with, we consider a general nonlinear dynamical system that is modeled by a set of ordinary differential equations (ODEs) in the following form:

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}) \tag{1}$$

with

$$\boldsymbol{y}(t) = \boldsymbol{h}[\boldsymbol{x}(t)]. \tag{2}$$

The state variable $\boldsymbol{x}(t) = [x_1(t), \ldots, x_n(t)]^\top \in \mathbb{R}^n$ in system (1) is of n-dimension, $\boldsymbol{y}(t) \in \mathbb{R}^p$ in Eq. (2) represents the output, $\boldsymbol{F}: \mathbb{R}^n \to \mathbb{R}^n$ stands for a continuous vector function, and $\boldsymbol{h}: \mathbb{R}^n \to \mathbb{R}^p$ with $p \leq n$ is a smooth observable function. In particular, if $\boldsymbol{h}$ is an identity function, $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \equiv \boldsymbol{x}$, which implies that each variable in system (1) is observable and measurable. However, if $p < n$ for $\boldsymbol{h}$, the observable function only contains information from a partial group or a combination of the components in the state variable $\boldsymbol{x}$. In the following discussions, we always suppose that there is a chaotic attractor, denoted by $\mathcal{A}$, contained in some bounded set $\Omega \subset \mathbb{R}^n$ in the phase

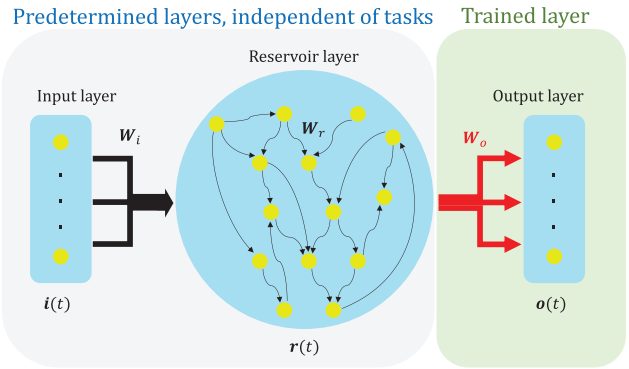

FIG. 1. A schematic illustration on the RC framework, including an input layer, a reservoir layer, an output layer, and the connection weights, W_i, W_o, and W_r, in between the different layers and among the nodes in the reservoir layer, respectively.

space. Our purpose is to detect the UPOs embedded in the attractor $\mathcal{A}$ without knowing the specific form of system (1) but with the observed data of time series.

First, we use the standard RC framework to restore the unknown chaotic system. To this end, we briefly introduce this framework, which actually consists of three typical parts:

1. an input layer with N_i input components, which is then transformed by a linear mapping,
2. a reservoir network with N_r neurons, which is driven by the transformed input signal, including itself, and then fed into a nonlinear function, and
3. an output layer with N_o output nodes, which linearly transforms the signal of the reservoir neurons.

The general idea behind is intuitively illustrated in Fig. 1. More specifically, an RC framework can be described as follows:

$$\boldsymbol{r}(t+\Delta t) = \tanh\left[\boldsymbol{W}_r\boldsymbol{r}(t) + \boldsymbol{W}_i\boldsymbol{\eta}(t)\right] \tag{3}$$

and

$$\boldsymbol{o}(t+\Delta t) = \boldsymbol{F}_o\left[\boldsymbol{r}(t+\Delta t)\right], \tag{4}$$

where $\boldsymbol{\eta}(t) = [\eta_1(t), \ldots, \eta_{N_i}(t)]^\top \in \mathbb{R}^{N_i}$ is the input signal, $\boldsymbol{r}(t) = [r_1(t), \ldots, r_{N_r}(t)]^\top \in \mathbb{R}^{N_r}$ is the reservoir state of the network with N_r reservoir nodes, $\boldsymbol{o}(t) = [o_1(t), \ldots, o_{N_o}(t)]^\top \in \mathbb{R}^{N_o}$ is the output, $\Delta t \ll 1$ represents the infinitesimal time step, $\boldsymbol{W}_i \in \mathbb{R}^{N_r \times N_i}$ is the linear input weight matrix, $\boldsymbol{W}_r$ is the weighted adjacency matrix of the reservoir layer, the vector function $\boldsymbol{F}_o[\cdot]$, which usually is taken as a polynomial mapping with a lower degree, maps the reservoir state $\boldsymbol{r}$ to the output $\boldsymbol{o}$, and $\tanh[\cdot]$ is the elementwise hyperbolic tangent function, also referred as an activation function.

As for the specific structure configurations of the RC, we make the following introductions. The matrices $\boldsymbol{W}_i$ and $\boldsymbol{W}_r$ are generated by a random mechanism, while the output function $\boldsymbol{F}_o[\cdot]$ is configured by some specific regression task. More precisely, the sth $(s = 1, 2, \ldots, N_i)$ component of the input signal is fed into the N_r/N_i neurons in the reservoir network through the input weight matrix $\boldsymbol{W}_i$, where the elements connecting the sth component with the fed neurons are sampled from a uniform distribution in $[-\sigma, \sigma]$, while the other elements are set as zero. For the network inside the RC, the weighted adjacency matrix $\boldsymbol{W}_r$ is constructed based on a sparse and large Erdös-Rényi random networks with a low average degree, denoted by d so that its element is sampled from a uniform distribution in $[0, \alpha]$, and then scaled by a positive scalar (i.e., multiplied by $\boldsymbol{W}_r$) so as to make its largest eigenvalue approximately approaching a preset parameter ρ, the spectral radius of the RC. Notably, once both the matrices $\boldsymbol{W}_i$ and $\boldsymbol{W}_r$ are initially randomly determined, they are no longer variant with the time evolution and the task selection. This also indicates the exact reason why the RC finds favor with the implementation in hardware designs.[39–41] Additionally, the traditional goal of the output layer is to train the RC for minimizing the difference, usually measured by the L_2 norm $\|\cdot\|$, between the output $\boldsymbol{o}(t)$ and the expected output $\boldsymbol{o}_e(t)$. In this article, the expected output (at time $t + \Delta t$) is the input signal (at time t) obtained from the continuous-time chaotic dynamical systems but with an infinitesimal shift along the time scale, and thus the goal in a mathematical sense of continuous time turns to minimize the difference $\|\boldsymbol{o}(t+\Delta t) - \boldsymbol{\eta}(t+\Delta t)\|$. Particularly, we choose the output function as follows:

$$\boldsymbol{F}_o\left[\boldsymbol{r}\right] = \boldsymbol{W}_{o_1}\boldsymbol{r} + \boldsymbol{W}_{o_2}\boldsymbol{r}^2,$$

where $\boldsymbol{W}_{o_1}, \boldsymbol{W}_{o_2} \in \mathbb{R}^{N_o \times N_r}$ and the components in the even (odd, resp.) columns of $\boldsymbol{W}_{o_1}$ ($\boldsymbol{W}_{o_2}$, resp.) are all zeros. In what follows, we describe the "training data" that we use in the RC framework. In order to have the data that can include sufficient information of the attractor and exclude the transition dynamics, we do not use the data in the interval $[-t_0\Delta t, 0]$ but use the data in $[0, T\Delta t]$ as the training data. Here, t_0 represents the step number of the transition duration and T is the step number of the training data. We thereby set the input signal sequence as $[\boldsymbol{\eta}(0), \boldsymbol{\eta}(\Delta t), \ldots, \boldsymbol{\eta}(T\Delta t)]$ and the driven reservoir state sequence as $[\boldsymbol{r}(0), \boldsymbol{r}(\Delta t), \ldots, \boldsymbol{r}(T\Delta t)]$. The practical mission of the above mathematical goal becomes an exploration of the output weights $\boldsymbol{W}_{o_1}$ and $\boldsymbol{W}_{o_2}$ via utilizing the Tikhonov regularized regression[43] for minimizing the loss function:

$$\begin{aligned}\mathcal{L} = &\sum_{j=0}^{T}\left\|\left[\boldsymbol{W}_{o_1}\boldsymbol{r}(j\Delta t) + \boldsymbol{W}_{o_2}\boldsymbol{r}^2(j\Delta t)\right] - \boldsymbol{\eta}(j\Delta t)\right\|^2 \\ &+ \beta\sum_{k=1}^{N_i}\left\|[\boldsymbol{W}_{o_1}]_k\right\|^2 + \left\|[\boldsymbol{W}_{o_2}]_k\right\|^2,\end{aligned} \tag{5}$$

where $[\boldsymbol{W}_{o_1}]_k$ ($[\boldsymbol{W}_{o_2}]_k$, resp.) represents the kth row of the matrix $\boldsymbol{W}_{o_1}$ ($\boldsymbol{W}_{o_2}$, resp.) and $\beta > 0$ is a small regularization coefficient penalizing large values of the fitting parameters (i.e., $\boldsymbol{W}_{o_1}$ and $\boldsymbol{W}_{o_2}$) to avoid over-fitting problems. The training time series $\boldsymbol{\eta}(t)$ and the reservoir evolution states $\boldsymbol{r}(t)$ for $t = 0, \Delta t, \ldots, T\Delta t$ are, respectively, stored in order as the columns of the matrices $\boldsymbol{\eta}$ and $\boldsymbol{R}$. Furthermore, $\boldsymbol{S}$ is an auxiliary matrix, resulting from the matrix $\boldsymbol{R}$ by squaring each component in its even rows. Consequently, the solution for the Tikhonov regularized regression in (5) is

$$\boldsymbol{W}_o = \boldsymbol{\eta}\boldsymbol{S}^\top\left(\boldsymbol{S}\boldsymbol{S}^\top + \beta\boldsymbol{E}\right)^{-1}. \tag{6}$$

Here, $\boldsymbol{E}$ is the identity matrix of a proper dimension and $\boldsymbol{W}_o$, whose odd (even, resp.) columns are the odd (even, resp.) columns of $\boldsymbol{W}_{o_1}$ ($\boldsymbol{W}_{o_2}$, resp.), is written as $\boldsymbol{W}_o = \boldsymbol{W}_{o_1} + \boldsymbol{W}_{o_2}$. The invertibility in

TABLE I. The parameters of the RC that we use in this article.

Parameters	Value	Parameters	Value
Δt	0.001	d	3.0
N_r	1200	T	5×10^5
β	10^{-4}	α	~0.27
ρ	0.6	σ	1.0

Eq. (6) is guaranteed by an appropriate selection of $\beta > 0$. It should be noted that the frame of the model, including the choices of parameters and the generation of random matrices, is configured in the almost same manner as those in Ref. 37; however, some parameters need fine-tuning in practice. Specifically, the parameters that we use in this article are introduced in Table I.

Using the RC framework introduced above and the training data that are supposed to be observed from the unknown dynamical system (1), we thus are able to approximate this system in the following discrete-time form:

$$\boldsymbol{\eta}(t+\Delta t) = \boldsymbol{F}_o\left[\tanh\left[\boldsymbol{W}_r\boldsymbol{r}(t) + \boldsymbol{W}_i\boldsymbol{\eta}(t)\right]\right], \tag{7}$$

where $\boldsymbol{\eta}$ is exactly the output $\boldsymbol{y}$ of the observable function $\boldsymbol{h}$ in (2) of system (1). More precisely, as $\Delta t \to 0$, the above discrete-time equation can be transformed as

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{\eta}(t)}{\mathrm{d}t} &= \lim_{\Delta t\to 0}\frac{\boldsymbol{\eta}(t+\Delta t)-\boldsymbol{\eta}(t)}{\Delta t}\\ &= \lim_{\Delta t\to 0}\frac{\boldsymbol{F}_o\left[\tanh\left[\boldsymbol{W}_r\boldsymbol{r}(t)+\boldsymbol{W}_i\boldsymbol{\eta}(t)\right]\right]-\boldsymbol{\eta}(t)}{\Delta t}.\end{aligned}$$

These become a group of differential equations in a closed form, which is akin to system (1). Practically, as the time increment is chosen as a pretty small value (i.e., $\Delta t = 10^{-3}$), $\{\boldsymbol{F}_o\left[\tanh\left[\boldsymbol{W}_r\boldsymbol{r}(t)+\boldsymbol{W}_i\boldsymbol{\eta}(t)\right]\right]-\boldsymbol{\eta}(t)\}/\Delta t$ is thereby regarded as a good approximation of the whole or partial vector field of system (1), that is

$$\widetilde{\boldsymbol{F}}(\boldsymbol{\eta}) \triangleq \frac{\boldsymbol{F}_o\left[\tanh\left[\boldsymbol{W}_r\boldsymbol{r}(t)+\boldsymbol{W}_i\boldsymbol{\eta}(t)\right]\right]-\boldsymbol{\eta}(t)}{\Delta t} \simeq \boldsymbol{F}(\boldsymbol{\eta}). \tag{8}$$

B. Importation of ADFC to RC

Now, with the approximated model (8) by the RC framework, we next aim to detect the UPOs by utilizing the ADFC technique developed in Refs. 6, 7, and 44. One advantage of such a method is no requirement of the detailed information of the UPOs prior to the control implementation. Precisely, applying the ADFC to the approximated model (8) yields

$$\dot{\boldsymbol{u}} = \widetilde{\boldsymbol{F}}(\boldsymbol{u}) + \boldsymbol{G}(t), \tag{9}$$

where $\boldsymbol{G}(t) = \Gamma(t)\{\boldsymbol{u}[t-\tau(t)] - \boldsymbol{u}(t)\}$ is the delayed feedback control term. For simplicity of control implementation, the delayed feedback term is only applied to one of the components, i.e., $\Gamma(t) = \mathrm{diag}\{0,\ldots,0,\gamma_k(t),0,\ldots 0\}$. Notably, the adaptiveness renders the time delay as well as the control gain time-variant, updating their dynamics in the following respective manners:

$$\begin{aligned}\dot{\tau} &= -r_1\left\{u_k[t-\tau(t)] - u_k(t)\right\},\\ \dot{\gamma}_k &= r_2\left\{u_k[t-\tau(t)] - u_k(t)\right\}^2,\end{aligned} \tag{10}$$

where r_1 and r_2 are proper positive constants as the free parameters to be fine-tuned to achieve a good convergence of the ADFC.

Since the controlled RC model (9) with the adaptive rules (10) is a type of the infinite-dimensional system, requiring initial values to be in function forms, we use sufficiently long trained data as the initial values for the controlled RC model and constant functions for the other variables associated with the ADFC. To guarantee the positiveness of the time delay and the periods in a concerned interval, we reset the value of τ to a small and positive value whenever it becomes zero or exceeds a preset value. Due to the monotonicity of γ_k in the adaptive rules (10), dynamics staying away from the UPOs provide a persist and pretty large control gain to more rapidly stabilize the UPOs embedded in the attractor. Once the stabilization is done, due to the noninvasiveness of the ADFC technique, the time-delay variable τ actually converges to a constant value (i.e., the period for one of the unknown UPOs).

As for the numerical simulations, we apply the traditional Euler algorithm to compute the controlled system (9), as well as the adaptive rules (10), with the step size Δt set above. Specifically, the discrete-time scheme for the controlled system (9) can be written as

$$\boldsymbol{u}(t+\Delta t) = \boldsymbol{F}_o\left[\tanh\left[\boldsymbol{W}_r\boldsymbol{r}(t)+\boldsymbol{W}_i\boldsymbol{u}(t)\right]\right] + \Delta t\cdot\boldsymbol{G}(t),$$

where $\tau(t)$ in $\boldsymbol{G}(t)$ is numerically updated according to the rule in (10) with the time unit Δt.

III. ILLUSTRATIVE EXAMPLES

To demonstrate the effectiveness of the method articulated in Sec. II, representative examples are provided, including the Lorenz system and the Mackey-Glass system. Additionally, utilizing the detected UPO and the pinning control strategy, we realize the synchronization among a group of the coupled Lorenz systems.

A. UPOs detection in the Lorenz system

First, we consider the Lorenz system,[45,46] a paradigm dynamical system, whose dynamics $\boldsymbol{x} = [x_1, x_2, x_3]^\top$ are described by the following ODEs:

$$\begin{aligned}\dot{x}_1 &= 10(x_2 - x_1),\\ \dot{x}_2 &= -10x_1x_3 + 28x_1 - x_2,\\ \dot{x}_3 &= 10x_1x_2 - 8x_3/3.\end{aligned} \tag{11}$$

As is well known, the Lorenz system (11) coins a chaotic attractor contained in the cube $\Phi = [-2,2]\times[-3,3]\times[0,5]$, where infinitely many and unobserved UPOs also exist.[47] The time series generated by this system used for training the RC are shown in Fig. 2(a), and a part of the time series for the reservoir state $\boldsymbol{r}$ is shown in Fig. 2(b). By operating the principal component analysis (PCA) on the time series of $\boldsymbol{r}$ as similarly done in Ref. 48, we display the first three principle components of $\boldsymbol{r}$ in Fig. 2(c). It indicates that, although the dimension of $\boldsymbol{r}$ is extremely high, the dynamics actually evolve in a low-dimensional manifold. Thus, after a completion of

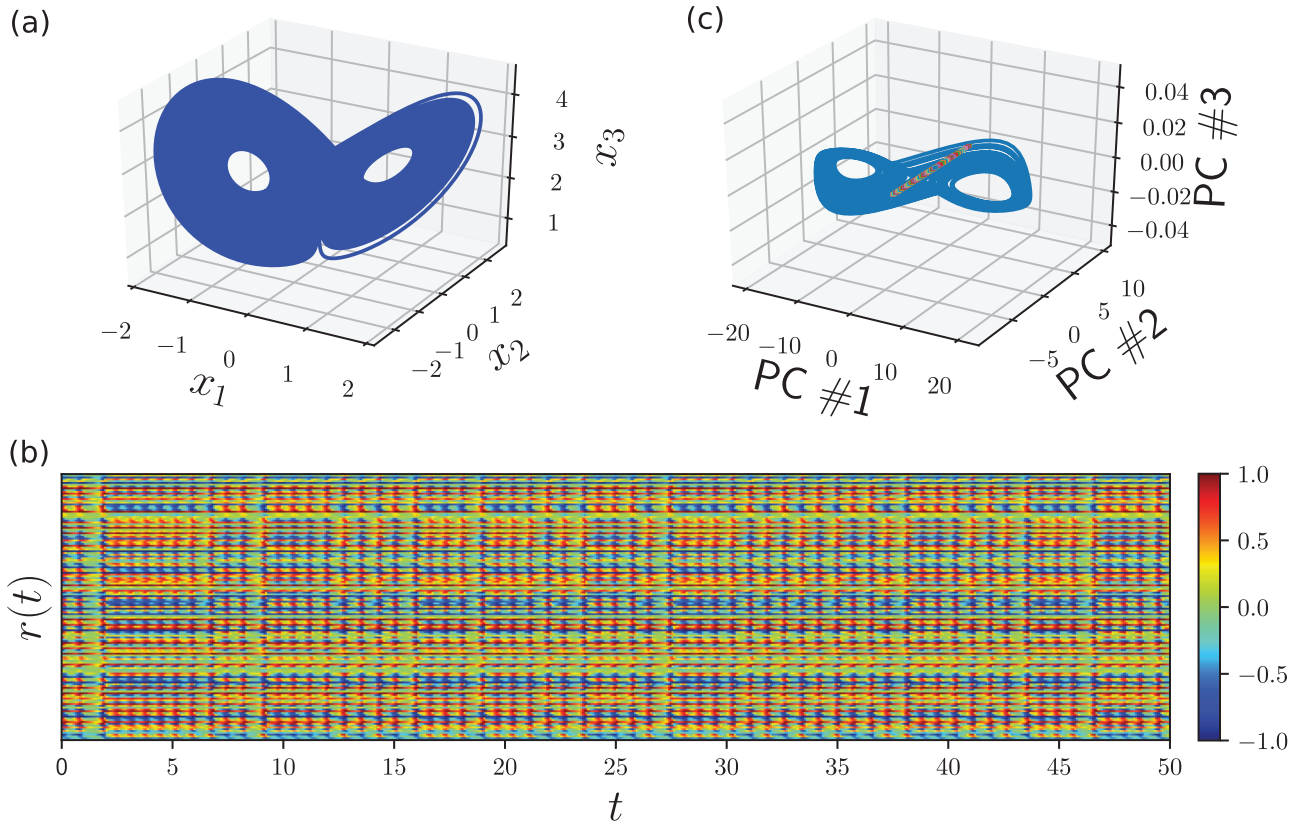

FIG. 2. (a) The time series of the Lorenz system (11) in the phase space, which are used for training the RC. The duration of the time series is from $t = 0$ to $t = 300$ and the time step is $\Delta t = 0.001$. (b) A part of the time series of $\boldsymbol{r}$ change with time after the training of the RC. (c) The dynamics of the first three principal components, $\{\text{PC \#}i\}_{i=1,2,3}$, in the phase space by operating the PCA method on $\boldsymbol{r}$ in (b).

the RC training, the input $\boldsymbol{\eta}$ with $\boldsymbol{r}$ forms a dynamical system, whose lower dimensional attractor is topologically equivalent to the original attractor of system (11).

Now, with the time series generated by system (11), we apply the above-articulated method to detect the UPOs. In particular, the controlled system (9) with the adaptive rules (10) for detecting UPOs in the Lorenz system (11) becomes

$$\begin{aligned} \dot{u}_1 &= \tilde{f}_1(\boldsymbol{u}), \\ \dot{u}_2 &= \tilde{f}_2(\boldsymbol{u}) + G(t), \\ \dot{u}_3 &= \tilde{f}_3(\boldsymbol{u}), \\ \dot{\tau} &= -r_1 \{u_2[t-\tau(t)] - u_2(t)\}, \\ \dot{\gamma} &= r_2 \{u_2[t-\tau(t)] - u_2(t)\}^2, \end{aligned} \tag{12}$$

where $\tilde{\boldsymbol{F}}(\boldsymbol{u}) = [\tilde{f}_1(\boldsymbol{u}), \tilde{f}_2(\boldsymbol{u}), \tilde{f}_3(\boldsymbol{u})]^\top$ is the approximated vector field by the RC framework, and the control is applied to the second variable as usual. Additionally, to ensure the small intensity of the perturbation G and the boundedness of the controlled system along the time, the configuration for G, as proposed in Refs. 42 and 44, is employed as

$$G(t) = \mathcal{I}_{\{|S(t)|<\bar{G}\}} S(t) + \bar{G}\mathcal{I}_{\{|S(t)|>\bar{G}\}} - \bar{G}\mathcal{I}_{\{|S(t)|<-\bar{G}\}},$$

where $S(t) = \gamma(t)\{x_2[t-\tau(t)] - x_2(t)\}$, $\mathcal{I}_C$ is the indication function of the set C, and $\bar{G} = 0.1$ can be an arbitrarily small constant, regarded as an empirical parameter. Through different initializations of the controlled system (12) and resetting $\tau(t)$ after every iteration of 2×10^4 time steps (totally 3 times for resetting), a few UPOs are founded according to the convergence of the time-delay variable τ. In particular, each identified period for the UPO is numerically determined by the average value of the variable τ on an interval where both the fluctuation variance of τ and the absolute value of the difference of τ's values at the beginning and the ending points are sufficiently small. For instance, the identified the periods as well as the shaded intervals for the variable τ in Fig. 3 are found by setting the critical values for the variance and the difference as 10^{-5} and 10^{-2}, respectively. Thus, several identified UPOs are presented numerically in Fig. 3, which are almost consistent with the typical UPOs reported in the literature.[49]

Next, we detect the UPOs for the case where $\boldsymbol{h}$ in (2) is not an identity function but satisfies $p < n$, i.e., only partial states can

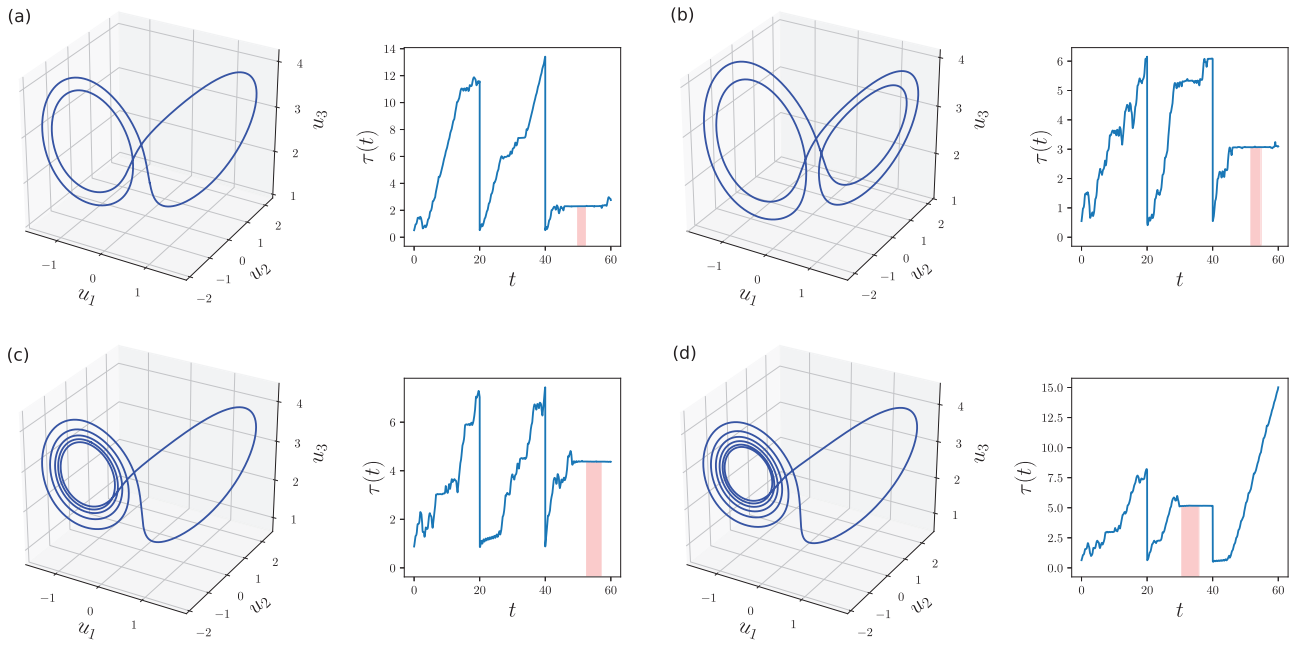

FIG. 3. The detected UPOs by using the articulated method (9) with (10). Each period of the UPO is determined by the average value of $\tau(t)$ on an interval (see the shaded interval in the right panel of each subfigure). The criteria for determining these periods as well as the intervals are introduced in the main text. Numerically, the detected periods are 2.3036 (a), 3.0700 (b), 4.3770 (c), and 5.1610 (d), respectively.

be observed. Particularly, assume that only the time series for $x_2(t)$ in system (11) is available. We do not use the delay-coordinate embedding technique to the time series directly. Instead, we use the following RC framework for fitting $x_2(t)$:

$$\begin{aligned} \boldsymbol{r}(t+\Delta t) = \tanh[&\boldsymbol{W}_0\boldsymbol{r}(t) + \boldsymbol{W}_1\boldsymbol{r}(t-\tau_{rc}) \\ &+ \boldsymbol{W}_2\boldsymbol{r}(t-2\tau_{rc}) + \boldsymbol{W}_i\eta(t)]. \end{aligned} \tag{13}$$

This framework is different from the elementary framework presented in (3) since the delays τ_{rc} and $2\tau_{rc}$ are taken into account for the reservoir state $\boldsymbol{r}$. Here, $x_2(t)$ is fed to the input $\eta(t)$ and the delay is taken as $\tau_{rc} = 0.1$. After training the RC model with delay, we have a partial vector field of the Lorenz system in an approximation form. We thus use the above-articulated control strategy to the approximated vector field as follows:

$$\begin{aligned} \dot{u}_2 &= \tilde{f}(u_2) + G(t), \\ \dot{\tau} &= -r_1\{u_2[t-\tau(t)] - u_2(t)\}, \\ \dot{\gamma} &= r_2\{u_2[t-\tau(t)] - u_2(t)\}^2, \end{aligned} \tag{14}$$

where

$$\tilde{f}(u_2) = \frac{u_2(t+\Delta t) - u_2(t)}{\Delta t} = \frac{\boldsymbol{F}_o[\boldsymbol{r}(t+\Delta t)] - u_2(t)}{\Delta t},$$

analogous to (8) for $\boldsymbol{\eta} = \boldsymbol{u}$. As shown in Fig. 4, a UPO with an approximated period, 3.7676, is successfully detected as the number of the reservoir nodes is chosen as 4.8×10^3 in the RC framework with delays.

Additionally, as shown in Fig. 5, the larger the number of the reservoir nodes, the more accurate dynamics the approximated model displays. Such a tendency does not matter with the introduction of delays into the RC framework. However, compared to the case where the delay-coordinate embedding technique is used directly to the signal by setting $\boldsymbol{\eta}(t) = [x_2(t), x_2(t-\tau_0), x_2(t-2\tau_0)]^\top$ for the RC in (3), the RC without using the embedding technique but with an introduction of delays as presented in (13) is able to achieve the most accurate restoration of the concerned dynamics. Such an accurate

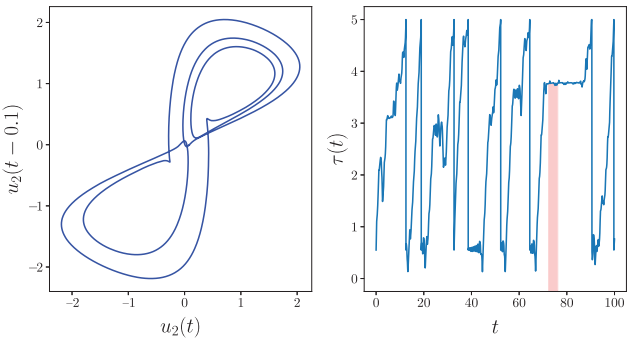

FIG. 4. A detected UPO when the time series of the variable x_2 of system (11) are available for training the RC with delays as proposed in (13). The period of the UPO is determined by the average value of τ along the shaded interval. This interval is chosen in the same manner as those in Fig. 3.

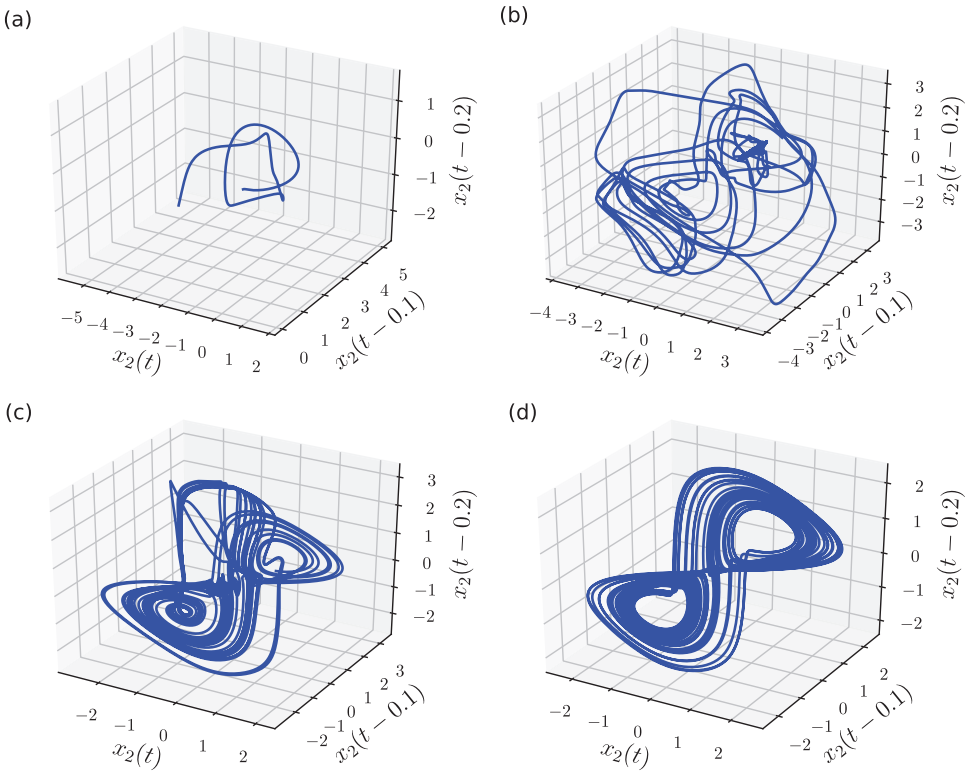

FIG. 5. Dynamics prediction of $x_2(t)$ in the three-dimensional phase space based on the model approximated by different configurations of the RC. (a) and (c) use the delay-coordinate embedding technique to the input signal of the RC by setting $\eta(t) = [x_2(t), x_2(t-0.1), x_2(t-0.2)]$ and also use the RC framework (3) containing different numbers of the reservoir nodes, 1200 and 4800, respectively. (b) and (d) do not use the embedding technique to the input signal but introduce $\tau_{rc} = 0.1$ into the RC framework as presented in (13), and still use different numbers of the reservoir nodes, 1200 and 4800, respectively. The most accurate restoration in (d) of the original dynamics is beneficial to the success of UPOs detection, as shown in Fig. 4.

restoration is tremendously crucial to a successful detection of UPOs. Either insufficient number of reservoir nodes or absence of delays in the RC framework always results in the failure of UPOs detection. Actually, the introduction of delays in the RC framework is more likely to form a system's space of high or infinite dimension, into which the original dynamics are more easily embedded.

B. UPOs detection in the Mackey-Glass system

In this subsection, we further validate the effectiveness of our method by finding the UPOs in the Mackey-Glass system, a time-delayed and infinite-dimensional model.[50] Such a system can be described by

$$\dot{x} = \frac{ax(t-\tau_m)}{1+x^b(t-\tau_m)} - cx, \tag{15}$$

where x describes the concentration of the circulating blood cells, τ_m is a constant feedback time delay, and a, b and c are the parameters of biological significance. For particular parameters and time delay, system (15) can coin a chaotic attractor, whose dimension increases as τ_m increases.[51] Figure 6(a) shows such an attractor, whose time series are used to train the RC with delay as follows:

$$\boldsymbol{r}(t+\Delta t) = \tanh\left[\boldsymbol{W}_0\boldsymbol{r}(t) + \boldsymbol{W}_1\boldsymbol{r}(t-\tau_{rc}) + \boldsymbol{W}_i\eta(t)\right], \tag{16}$$

where $\eta(t) = x(t)$. Now, using the above-articulated method to the model approximated by (16), we are able to detect a UPO. As shown in Figs. 6(b) and 6(c), such a detected UPO by using our model-free method is almost the same as the one that is detected by using the ADFC directly to the original system (15).

Akin to the case where only one variable of the Lorenz system is observable, importing the ADFC to the model approximated by the RC framework (3) without delay cannot easily detect the UPOs of the Mackey-Glass model, as shown in Fig. 6(d). In fact, although the approximated models trained by the RC with delay [Fig. 6(e)] and without delay [see Fig. 6(f)] can both fit the training data very well, their prediction performances are different. Particularly, as shown in Figs. 6(d) and 6(f), the RC framework (16) with $\tau_{rc} = \tau_m$ makes the approximated model having strong predictability, which therefore renders the success of the UPO detection in Fig. 6(c). These investigations also indicate that a successful or failed detection of the UPO can be regarded as a criterion for how accurately the model is approximated by the RC framework.

C. Synchronizing the coupled Lorenz systems to the detected UPO

The UPOs of nonlinear systems are observable with probability zero and locating them usually requires highly computational cost. This thus makes control/synchronization problems becoming extremely challenging, since solving these problems usually needs *a priori* knowledge of the control target. Thus, the method articulated above provides a way to locate the UPO approximately and then realize the control/synchronization. To validate this point, we use the UPO identified in the above example of the Lorenz system as the control/synchronization target, and also use the pinning control strategy,

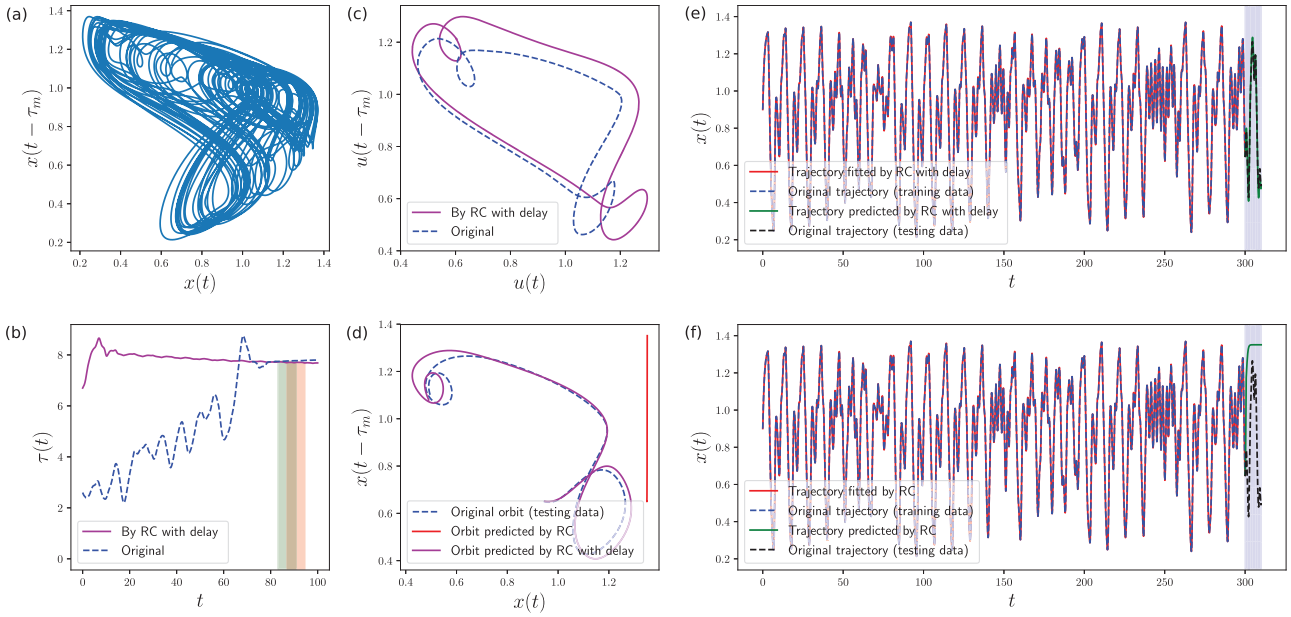

FIG. 6. (a) The chaotic attractor coined by the Mackey-Glass system (15) with $a = 2$, $b = 10$, $c = 1$, and $\tau_m = 3.18$. The variations of τ with the time t (b) and the correspondingly detected UPOs (c) for using the ADFC to, respectively, the original model (15) and the model approximated by the RC with delay (16). Here, the periods of the UPOs are estimated by the average values of τ on the shaded intervals, that is, 7.76 for the original model and 7.70 for the approximated model. The delay in the RC is set as $\tau_{rc} = \tau_m$. The shaded intervals in (b) are determined in the same manner as those in Fig. 4. The RC framework with delay after training shows stronger predictability, as shown by the solid and twisty orbit in the phase plane (d) as well as by the solid trajectory in the shaded region along the time evolution (e). For the RC framework with and without delay, the fitting accuracies on the training data [i.e., the trajectories along the unshaded intervals in (e) and (f)] approach a satisfactorily high level; however, the prediction accuracies on the testing data are in completely different manners along the shaded intervals in (e) and (f).

as developed in Ref. 52, to realize synchronization to a periodic orbit among the following coupled Lorenz systems:

$$\begin{aligned}\dot{\boldsymbol{u}}^{i^*} &= \boldsymbol{L}(\boldsymbol{u}^{i^*}) + C\sum_{j=1}^{q} a_{i^*j}\boldsymbol{u}^j(t) - C\varepsilon\left[\boldsymbol{u}^{i^*} - \boldsymbol{u}(t)\right],\\ \dot{\boldsymbol{u}}^{i} &= \boldsymbol{L}(\boldsymbol{u}^{i}) + C\sum_{j=1}^{q} a_{ij}\boldsymbol{u}^j(t), \quad i = 1,\ldots,q \text{ and } i \neq i^*,\end{aligned} \tag{17}$$

where the ith node $\boldsymbol{u}^i \in \mathbb{R}^3$, the vector field $\boldsymbol{L}(\cdot)$ satisfies the uncontrolled Lorenz system (11), the control/synchronization target $\boldsymbol{u}(t)$ is the UPO with the period 2.3036 approximately detected in Fig. 3(a), $\varepsilon = 10$ is the control gain, $C = 10$ is the coupling strength, $i^* = 3$ is the index of the controlled node in the network whose size is set as $q = 6$, and

$$\boldsymbol{A} = (a_{ij})_{q\times q} = \begin{pmatrix} -3.44 & 0.90 & 0.40 & 0.62 & 0.64 & 0.88\\ 0.90 & -2.81 & 0.30 & 0. & 0.70 & 0.9\\ 0.39 & 0.299 & -1.98 & 0.88 & 0. & 0.41\\ 0.62 & 0. & 0.88 & -1.96 & 0. & 0.45\\ 0.64 & 0.70 & 0. & 0. & -1.61 & 0.27\\ 0.88 & 0.90 & 0.41 & 0.45 & 0.27 & -2.91 \end{pmatrix}$$

is the weighted adjacency matrix. As numerically validated in Fig. 7, the coupled systems, starting from different initial values, synchronize rapidly to the unknown but estimated UPO. This definitely confirms the detection accuracy of the UPO detected by our articulated model-free method.

IV. INTERPRETATION OF RC: A VIEWPOINT OF SYNCHRONIZATION

In this section, we illustrate how the RC works effectively in a sense of synchronization. To this end, we first write the dynamical system (1) approximately in a discrete-time form

$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}(t) + \Delta t \cdot \boldsymbol{F}[\boldsymbol{x}(t)]. \tag{18}$$

Substituting the time series $\boldsymbol{x}(t)$ for $\boldsymbol{\eta}(t)$ in the RC framework (3) yields

$$\boldsymbol{r}(t + \Delta t) = \tanh\left[\boldsymbol{W}_r\boldsymbol{r}(t) + \boldsymbol{W}_i\boldsymbol{x}(t)\right]. \tag{19}$$

Thus, training the RC successfully with $\boldsymbol{x}(t)$ suggests an realization of the synchronization between $\boldsymbol{x}(t)$ and $\boldsymbol{\eta}(t)$ in (3). Hence, to approach the synchronization, we add a coupling term into the output (7) of the RC framework as follows:

$$\boldsymbol{\eta}(t + \Delta t) = \boldsymbol{F}_o\left[\tanh\left[\boldsymbol{W}_r\boldsymbol{r}(t) + \boldsymbol{W}_i\boldsymbol{\eta}(t)\right]\right] - k[\boldsymbol{\eta}(t) - \boldsymbol{x}(t)], \tag{20}$$

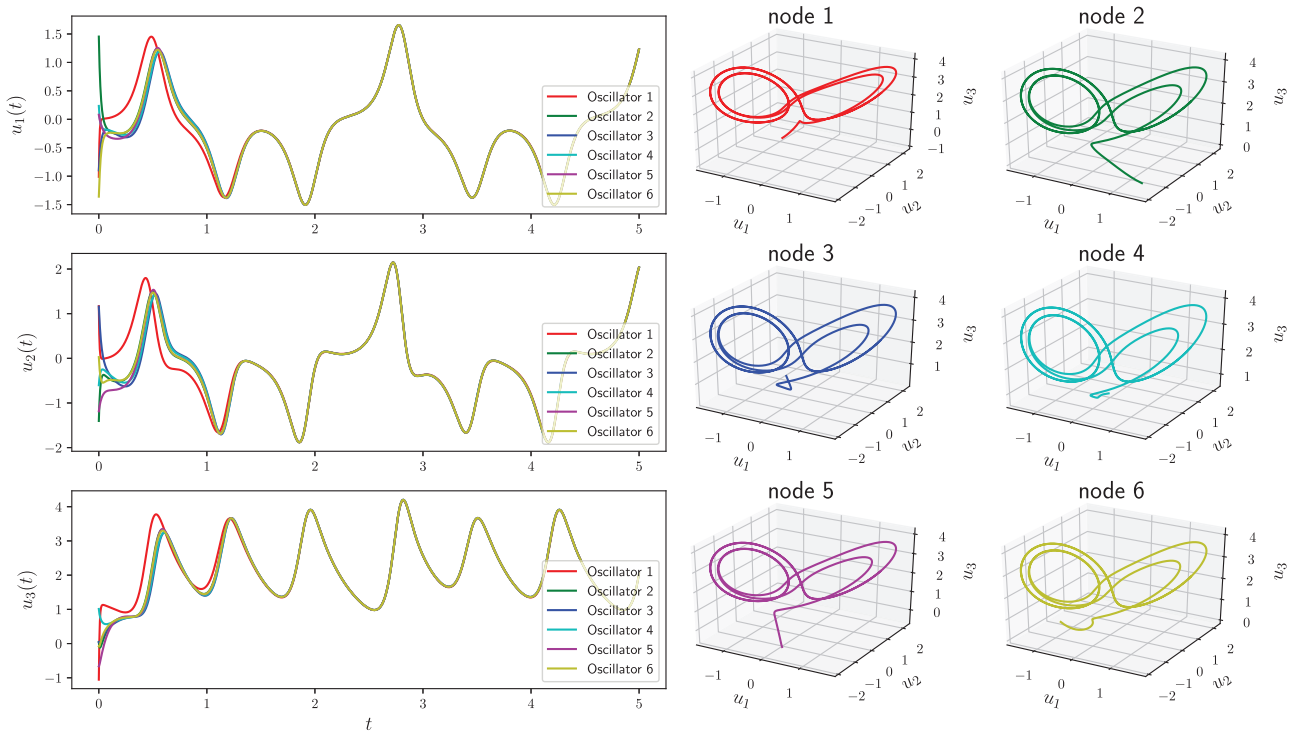

FIG. 7. Pinning control leads the coupled Lorenz systems (17), starting from different initial values, to a rapid synchronization of the UPO with the period 2.3036 approximately detected by our articulated method. Here, the UPO, as an unknown synchronization target, is numerically estimated in Fig. 3(a).

where the connecting matrices $\boldsymbol{W}_{r,i}$ and the output function $\boldsymbol{F}_o[\cdot]$ are generated as those in Sec. II, and the constant k is a coupling strength pending for selection.

Now, the above three equations, Eqs. (18), (19), and (20) together, form a discrete-time dynamical system and the error dynamics between $\boldsymbol{x}(t)$ and $\boldsymbol{\eta}(t)$ becomes

$$\begin{aligned}
&\boldsymbol{\eta}(t+\Delta t)-\boldsymbol{x}(t+\Delta t)\\
&\quad=\left\{\int_0^1 \partial_{\boldsymbol{x}}\widehat{\boldsymbol{F}}_o[\boldsymbol{r}(t),\boldsymbol{x}(t)+\theta[\boldsymbol{\eta}(t)-\boldsymbol{x}(t)]]\mathrm{d}\theta-k\cdot\boldsymbol{E}\right\}\\
&\qquad\times[\boldsymbol{\eta}(t)-\boldsymbol{x}(t)]+\left\{\widehat{\boldsymbol{F}}_o[\boldsymbol{r}(t),\boldsymbol{x}(t)]-\boldsymbol{x}(t+\Delta t)\right\}\\
&\quad\triangleq\boldsymbol{J}(t)\cdot[\boldsymbol{\eta}(t)-\boldsymbol{x}(t)]+\boldsymbol{\epsilon}(t),
\end{aligned}\tag{21}$$

where $\widehat{\boldsymbol{F}}_o[\boldsymbol{r}(t),\boldsymbol{\eta}(t)]\triangleq\boldsymbol{F}_o[\tanh[\boldsymbol{W}_r\boldsymbol{r}(t)+\boldsymbol{W}_i\boldsymbol{\eta}(t)]]$, $\partial_{\boldsymbol{x}}$ represents the partial derivative with respect to the argument $\boldsymbol{x}$, and $\boldsymbol{E}$ is the identity matrix as defined in Sec. II. Denote by $|\lambda|_{\max}(t)$ the maximal norm of the eigenvalue of the matrix $\boldsymbol{J}(t)$, and set $\epsilon(t)\triangleq\|\boldsymbol{\epsilon}(t)\|$. Analytically, to achieve the above-mentioned synchronization as well as the asymptotical stability of the error dynamics in (21), we require at least the validity of the following two conditions:

$$|\lambda|_{\max}(t)<\rho\quad\text{and}\quad\epsilon(t)\ll 1\tag{22}$$

for a constant $\rho<1$ and all sufficiently large t.

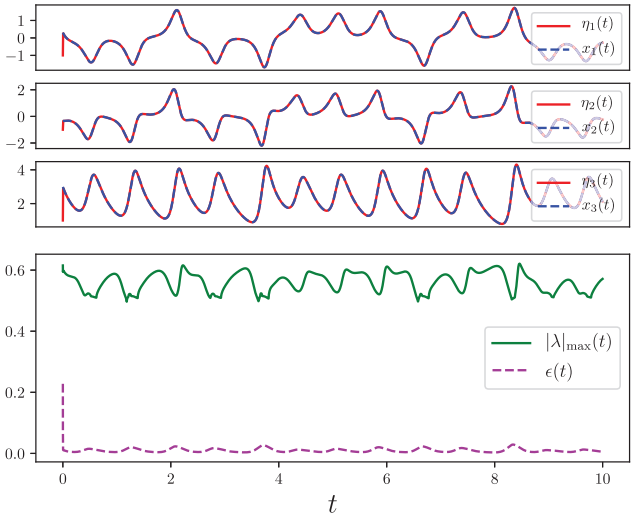

FIG. 8. A viewpoint of synchronization for effectiveness of the RC framework in dynamical systems learning and prediction. In the upper panel, $\boldsymbol{\eta}(t)$ in the RC framework (20) synchronizes very well with the training data $\boldsymbol{x}(t)$ produced by system (18) and, in the bottom panel, the two conditions in (22) are numerically validated. Here, for system (18), the Lorenz system (11) is used as an example.

In order to support the above arguments, we take the Lorenz system (11) with $k = 1/2$ as an example. Indeed, in Fig. 8, we numerically show a perfect synchronization between $\boldsymbol{\eta}(t)$ of the RC and the training data $\boldsymbol{x}(t)$ that are observed from the Lorenz system, and validate the conditions established in (22) as well. This, thus, provides a viewpoint of synchronization for illustrating how the RC works effectively for dynamical systems learning and prediction based only on the data of time series.

Additionally, for applying an RC framework successfully to time series prediction, although the number of the states of the RC dynamics in (19) is extremely large, the systems (18) and (19), as a whole, are actually evolving in a manifold of relatively low dimension. Thus, in light of the traditional and generalized coordinate embedding techniques, randomly picking up only a few but sufficient number of the RC states in (19) even without time delays, akin to the method articulated in Ref. 53, makes it achievable to reconstruct the dynamics of system (18) almost surely. Therefore, this, with the synchronization results presented above, can be seen as an analytical foundation for further and only using the output layer, as proposed in the original RC framework, to approximately learn and synchronize the data of time series from the system (18).

V. CONCLUDING REMARKS AND PERSPECTIVE

In this article, we have articulated a data-based and model-free method to locate UPOs embedded in chaotic strange attractors only using time series. Our approach actually integrates a well-known machine learning technique, the RC framework, with a complex system control strategy, the ADFC technique. Through using representative examples, we have presented how the configurations of the RC framework as well as the ADFC technique influence the accuracy of the UPOs detection. Additionally and more interestingly, we have provided an analytical argument from a viewpoint of synchronization for illustrating how the RC works effectively for dynamical systems learning and prediction based only on the data of time series. We expect that our approach as well as those numerical and analytical illustrations could become one of the contributions that connect machine learning with identification and modulation of complex systems.

As for the future research, the following three directions are suggested:

(i) some other typical structures, such as Refs. 39, 54, and 55, can be introduced for the RC framework to improve the performance of time series processing and UPOs detection,
(ii) paralleled reservoirs, such as the scheme proposed in Ref. 37, can be integrated into the current method for reducing the computational cost and detecting more UPOs that act as the skeleton of the chaotic attractor, and
(iii) adaptive scheme from the viewpoint of synchronization can be analytically and numerically developed for updating the weights of the randomly and sparsely connected neurons in the output layer of the RC framework, completely revealing its mechanism in dynamical system learning and prediction.

All these constitute our present or/and future research topics.

ACKNOWLEDGMENTS

W.L. was supported by the Research Institute of Intelligent and Complex Systems, Fudan University, China, the National Science Foundation of China (NSFC) (Grant Nos. 11322111 and 61773125), the National Key R&D Program of China (Grant No. 2018YFC0116600), and the Science and Technology Commission of Shanghai Municipality (Grant No. 18DZ1201000). H.M. was supported by the NSFC (Grant No. 11771010). Q.Z. was sponsored by the China Scholarship Council as a visiting graduate student at the Harvard Medical School, USA.

REFERENCES

[1] E. Ott, C. Grebogi, and J. Yorke, "Controlling chaos," Phys. Rev. Lett. **64**(11), 1196 (1990).
[2] P. Cvitanović, "Periodic orbits as the skeleton of classical and quantum chaos," Physica D **51**(1–3), 138–151 (1991).
[3] K. Pyragas, "Continuous control of chaos by self-controlling feedback," Phys. Lett. A **170**(6), 421–428 (1992).
[4] G. Chen and X. Yu, *Chaos Control: Theory and Applications* (Springer, Berlin, 2003).
[5] D. Huang, "Stabilizing near-nonhyperbolic chaotic systems with applications," Phys. Rev. Lett. **93**(21), 214101 (2004).
[6] W. Lin and H. Ma, "Failure of parameter identification based on adaptive synchronization techniques," Phys. Rev. E **75**(6), 066212 (2007).
[7] H. Ma and W. Lin, "Nonlinear adaptive synchronization rule for identification of a large amount of parameters in dynamical models," Phys. Lett. A **374**(2), 161–168 (2009).
[8] W. Lin and H. Ma, "Synchronization between adaptively coupled systems with discrete and distributed time-delays," IEEE Trans. Autom. Control **55**(4), 819–830 (2010).
[9] T. Ge, X. Tian, J. Kurths, J. Feng, and W. Lin, "Achieving modulated oscillations by feedback control," Phys. Rev. E **90**, 022909 (2014).
[10] S. Zhou, P. Ji, Q. Zhou, J. Feng, J. Kurths, and W. Lin, "Adaptive elimination of synchronization in coupled oscillator," New J. Phys. **19**, 083004 (2017).
[11] W. Lin, X. Chen, and S. Zhou, "Achieving control and synchronization merely through a stochastically adaptive feedback coupling," Chaos **27**, 073110 (2017).
[12] K. Pyragas, "Control of chaos via extended delay feedback," Phys. Lett. A **206**(5–6), 323–330 (1995).
[13] M. Gutzwiller, *Chaos in Classical and Quantum Mechanics* (Springer, Berlin, 2013), Vol. 1.
[14] D. Auerbach, P. Cvitanović, J. Eckmann, G. Gunaratne, and I. Procaccia, "Exploring chaotic motion through periodic orbits," Phys. Rev. Lett. **58**(23), 2387 (1987).
[15] D. Lathrop and E. Kostelich, "Characterization of an experimental strange attractor by periodic orbits," Phys. Lett. A **40**(7), 4028 (1989).
[16] G. Gunaratne, P. Linsay, and M. Vinson, "Chaos beyond onset: A comparison of theory and experiment," Phys. Rev. Lett. **63**(1), 1 (1989).
[17] P. So, S. J. Schiff, D. T. Kaplan, E. Ott, T. Sauer, and C. Grebogi, "Detecting unstable periodic orbits in experimental data," Phys. Rev. Lett. **76**(25), 4705 (1996).
[18] P. So, E. Ott, T. Sauer, B. J. Gluckman, C. Grebogi, and S. J. Schiff, "Extracting unstable periodic orbits from chaotic time series data," Phys. Rev. E **55**(5), 5398 (1997).
[19] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence* (Springer, 1981), pp. 366–381.
[20] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2012), pp. 1097–1105.
[21] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, N. Patrick, N. S. Tara *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Process. Mag. **29**(6), 82–97 (2012).
[22] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering

the game of go with deep neural networks and tree search," Nature **529**(7587), 484 (2016).
[23]R. Service, "Google's deepmind aces protein folding," Science (2018).
[24]E. J. Candés, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," Commun. Pure Appl. Math. **59**, 1207 (2006).
[25]*Compressed Sensing: Theory and Applications*, edited by Y. C. Eldar and G. Kutyniok (Cambridge University Press, 2012).
[26]W.-X. Wang, Y. C. Lai, and C. Grebogi, "Data based identification and prediction of nonlinear and complex dynamical systems," Phys. Rep. **644**(1), 1 (2016).
[27]H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," German Natl. Res. Center Information Technol. **148**(34), 13 (2001).
[28]W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," Neural Comput. **14**(11), 2531–2560 (2002).
[29]H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," Science **304**(5667), 78–80 (2004).
[30]H. Jaeger, "Echo state network," Scholarpedia **2**(9), 2330 (2007).
[31]M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," Comput. Sci. Rev. **3**(3), 127–149 (2009).
[32]W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi, "Predicting catastrophes in nonlinear dynamical systems by compressive sensing," Phys. Rev. Lett. **106**, 154101 (2011).
[33]X. Ji, Y. Wu, W. Sheng, and W. Lin, "Identification of interactions in fractional-order systems with high dimensions," Chaos **24**, 023119 (2014).
[34]J. Pathak, Z. Lu, B. Hunt, M. Girvan, and E. Ott, "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data," Chaos **27**(12), 121102 (2017).
[35]Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, "Reservoir observers: Model-free inference of unmeasured variables in chaotic systems," Chaos **27**(4), 041102 (2017).
[36]Z. Lu, B. Hunt, and E. Ott, "Attractor reconstruction by machine learning," Chaos **28**(6), 061104 (2018).
[37]J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," Phys. Rev. Lett. **120**(2), 024102 (2018).
[38]B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: Theory, applications and implementations," in *Proceedings of the 15th European Symposium on Artificial Neural Networks* (UGent, 2007), pp. 471–482.
[39]L. Appeltant, M. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," Nat. Commun. **2**, 468 (2011).
[40]L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. Udaltsov, Y. Chembo, and M. Jacquot, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," Phys. Rev. X **7**(1), 011015 (2017).
[41]F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," Opt. Express **20**(20), 22783–22795 (2012).
[42]H. Ma, W. Lin, and Y. Lai, "Detecting unstable periodic orbits in high-dimensional chaotic systems from time series: Reconstruction meeting with adaptation," Phys. Rev. E **87**(5), 050901 (2013).
[43]A. Tikhonov, V. Arsenin, and F. John, *Solutions of Ill-Posed Problems* (Winston, Washington, DC, 1977), Vol. 14.
[44]W. Lin, H. Ma, J. Feng, and G. Chen, "Locating unstable periodic orbits: When adaptation integrates into delayed feedback control," Phys. Rev. E **82**(4), 046214 (2010).
[45]E. Lorenz, "Deterministic nonperiodic flow," J. Atmos. Sci. **20**(2), 130–141 (1963).
[46]C. Sparrow, *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors* (Springer, New York, 1982).
[47]B. Boghosian, A. Brown, J. Lätt, H. Tang, L. Fazendeiro, and P. Coveney, "Unstable periodic orbits in the Lorenz attractor," Philos. Trans. R. Soc. A **369**(1944), 2345–2353 (2011).
[48]Z. Lu and D. Bassett, "A parsimonious dynamical model for structural learning in the human brain," e-print arXiv:1807.05214 (2018).
[49]Z. Galias and W. Tucker, "Short periodic orbits for the Lorenz system," in *International Conference on Signals and Electronic Systems* (IEEE, 2008), pp. 285–288.
[50]M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," Science **197**(4300), 287–289 (1977).
[51]J. Farmer, "Chaotic attractors of an infinite-dimensional dynamical system," Physica D **4**(3), 366–393 (1982).
[52]T. Chen, X. Liu, and W. Lu, "Pinning complex networks by a single controller," IEEE Trans. Circuits Syst. I **54**(6), 1317–1326 (2007).
[53]H. Ma, S. Leng, K. Aihara, W. Lin, and L. Chen, "Randomly distributed embedding making short-term high-dimensional data predictable," Proc. Natl. Acad. Sci. U.S.A. **115**(43), E9994–E10002 (2018).
[54]N. Haynes, M. Soriano, D. Rosin, I. Fischer, and D. Gauthier, "Reservoir computing with a single time-delay autonomous Boolean node," Phys. Rev. E **91**(2), 020801 (2015).
[55]T. Carroll, "Using reservoir computers to distinguish chaotic signals," Phys. Rev. E **98**(5), 052209 (2018).